Spring Data JPA系列：应用查询提示（Applying query hints）

这一节讲应用查询提示，学这一节比较波折，文档上的介绍太简单了，而且面对示例不知道如何下手，所以拖了一下，才有点头绪。

```
public interface UserRepository extends Repository&lt;User, Long&

  @QueryHints(value = { @QueryHint(name = "name", value = "value"
              forCounting = false)
  Page&lt;User&gt; findByLastname(String lastname, Pageable pagea
}
```

对name和value的值我一直以为是自定义的，最后发现不完全是，是有一组定义好的值供我们

1、QueryHints源码如下：

```
/*
 * Hibernate, Relational Persistence for Idiomatic Java
 *
 * License: GNU Lesser General Public License (LGPL), version 2.1
 * See the lgpl.txt file in the root directory or &lt;http://www.
 */
package org.hibernate.jpa;

import java.util.HashSet;
import java.util.Set;

import static org.hibernate.annotations.QueryHints.CACHEABLE;
import static org.hibernate.annotations.QueryHints.CACHE_MODE;
import static org.hibernate.annotations.QueryHints.CACHE_REGION;
import static org.hibernate.annotations.QueryHints.COMMENT;
import static org.hibernate.annotations.QueryHints.FETCHGRAPH;
import static org.hibernate.annotations.QueryHints.FETCH_SIZE;
import static org.hibernate.annotations.QueryHints.FLUSH_MODE;
import static org.hibernate.annotations.QueryHints.LOADGRAPH;
import static org.hibernate.annotations.QueryHints.NATIVE_LOCKMOD
import static org.hibernate.annotations.QueryHints.READ_ONLY;
import static org.hibernate.annotations.QueryHints.TIMEOUT_HIBERN
import static org.hibernate.annotations.QueryHints.TIMEOUT_JPA;

/**
 * Defines the supported JPA query hints
 *
 * @author Steve Ebersole
 */
public class QueryHints {
    /**
     * The hint key for specifying a query timeout per Hibernate
     *
     * @deprecated use {@link #SPEC_HINT_TIMEOUT} instead
     */
    @Deprecated
    public static final String HINT_TIMEOUT = TIMEOUT_HIBERNATE;

    /**
     * The hint key for specifying a query timeout per JPA, which
     */
```

Spring Data

阅读	评论	喜欢
261	0	0

喜欢

SpringForAll

文章	问答	粉丝
79	6	236

关注

```
public static final String SPEC_HINT_TIMEOUT = TIMEOUT_JPA;

/**
 * The hint key for specifying a comment which is to be embed
 */
public static final String HINT_COMMENT = COMMENT;

/**
 * The hint key for specifying a JDBC fetch size, used when e
 */
public static final String HINT_FETCH_SIZE = FETCH_SIZE;

/**
 * The hint key for specifying whether the query results shou
 * "same query".
 */
public static final String HINT_CACHEABLE = CACHEABLE;

/**
 * The hint key for specifying the name of the cache region (
 * to use for storing the query results.
 */
public static final String HINT_CACHE_REGION = CACHE_REGION;

/**
 * The hint key for specifying that objects loaded into the p
 * should be associated with the persistence context as read-
 */
public static final String HINT_READONLY = READ_ONLY;

/**
 * The hint key for specifying the cache mode ({@link org.hib
 * execution of the hinted query.
 */
public static final String HINT_CACHE_MODE = CACHE_MODE;

/**
 * The hint key for specifying the flush mode ({@link org.hib
 * execution of the hinted query.
 */
public static final String HINT_FLUSH_MODE = FLUSH_MODE;

public static final String HINT_NATIVE_LOCKMODE = NATIVE_LOCK

/**
 * Hint providing a "fetchgraph" EntityGraph.  Attributes exp
 * FetchType.EAGER (via join fetch or subsequent select).
 *
 * Note: Currently, attributes that are not specified are tre
 * on the attribute's definition in metadata, rather than for
 */
public static final String HINT_FETCHGRAPH = FETCHGRAPH;

/**
 * Hint providing a "loadgraph" EntityGraph.  Attributes expl
 * FetchType.EAGER (via join fetch or subsequent select).  At
 * FetchType.LAZY or FetchType.EAGER depending on the attribu
 */
public static final String HINT_LOADGRAPH = LOADGRAPH;

private static final Set&lt;String&gt; HINTS = buildHintsSet(
```

```
        private static Set&lt;String&gt; buildHintsSet() {
            HashSet&lt;String&gt; hints = new HashSet&lt;String&gt;()
            hints.add( HINT_TIMEOUT );
            hints.add( SPEC_HINT_TIMEOUT );
            hints.add( HINT_COMMENT );
            hints.add( HINT_FETCH_SIZE );
            hints.add( HINT_CACHE_REGION );
            hints.add( HINT_CACHEABLE );
            hints.add( HINT_READONLY );
            hints.add( HINT_CACHE_MODE );
            hints.add( HINT_FLUSH_MODE );
            hints.add( HINT_NATIVE_LOCKMODE );
            hints.add( HINT_FETCHGRAPH );
            hints.add( HINT_LOADGRAPH );
            return java.util.Collections.unmodifiableSet( hints );
        }

        public static Set&lt;String&gt; getDefinedHints() {
            return HINTS;
        }

        protected QueryHints() {
        }
    }
```

对应值的介绍，除了注释，我在网上也找到了一些资料，讲的很详细：
https://www.thoughts-on-java.org/11-jpa-hibernate-query-hints-every-developer-know
https://docs.jboss.org/hibernate/orm/4.3/javadocs/org/hibernate/jpa/QueryHints.html

2、在CustomerRepository中添加示例：

导入QueryHint的name的对应的值

```
import static org.hibernate.jpa.QueryHints.HINT_COMMENT;

        /**
         * 一个参数，匹配两个字段
         * @param name2
         * @Param pageable 分页参数
         * @return
         * 这里Param的值和=:后面的参数匹配，但不需要和方法名对应的参数值对应
         * 这里增加了@QueryHints注解，是给查询添加一些额外的提示
         * 比如当前的name值为HINT_COMMENT是在查询的时候带上一些备注信息
         */
        @QueryHints(value = { @QueryHint(name = HINT_COMMENT, value =
        @Query("select c from Customer c where c.firstName=:name or c
        Page&lt;Customer&gt; findByName(@Param("name") String name2,P
```

3、在CustomerController中添加示例：

```
        /**
         * 分页
         * 应用查询提示@QueryHints，这里是在查询的适合增加了一个comment
         * 查询结果是lastName和firstName都是bauer这个值的数据
         */
        @RequestMapping("/pageable")
        public void pageable(){
            //Pageable是接口，PageRequest是接口实现
            //PageRequest的对象构造函数有多个，page是页数，初始值是0，size是
            Pageable pageable = new PageRequest(0,3, Sort.Direction.D
            Page&lt;Customer&gt; page = repository.findByName("bauer"
            //查询结果总行数
            System.out.println(page.getTotalElements());
            //按照当前分页大小，总页数
            System.out.println(page.getTotalPages());
            //按照当前页数、分页大小，查出的分页结果集合
            for (Customer customer: page.getContent()) {
                System.out.println(customer.toString());
            }
            System.out.println("-------------------------------------
        }
```

注意到，这里除了方法调用了带有查询提示的方法以外，还对方法的Pageable参数进行了简单实现——PageRequest，这个类包含了多个构造函数，可以根据自己的需求自由定制，对于排序不分参考Sort那一篇。

参考：
官方文档:https://docs.spring.io/spring-data/jpa/docs/current/reference/html
API官方文档:http://docs.spring.io/spring-data/data-jpa/docs/current/api/
JPQL文档:http://www.blogjava.net/calmJava/archive/2011/04/01/347450.html

DEMO示例：https://github.com/icnws/spring-data-jpa-demo

附：PageRequest源码：

```
/*
 * Copyright 2008-2014 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License")
 * you may not use this file except in compliance with the Licens
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, sof
 * distributed under the License is distributed on an "AS IS" BAS
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express o
 * See the License for the specific language governing permission
 * limitations under the License.
 */
package org.springframework.data.domain;

import org.springframework.data.domain.Sort.Direction;

/**
 * Basic Java Bean implementation of {@code Pageable}.
 *
 * @author Oliver Gierke
 * @author Thomas Darimont
 */
public class PageRequest extends AbstractPageRequest {

    private static final long serialVersionUID = -454150993895608

    private final Sort sort;

    /**
     * Creates a new {@link PageRequest}. Pages are zero indexed,
     * page.
     *
     * @param page zero-based page index.
     * @param size the size of the page to be returned.
     */
    public PageRequest(int page, int size) {
        this(page, size, null);
    }
```

```
/**
 * Creates a new {@link PageRequest} with sort parameters app
 *
 * @param page zero-based page index.
 * @param size the size of the page to be returned.
 * @param direction the direction of the {@link Sort} to be s
 * @param properties the properties to sort by, must not be {
 */
public PageRequest(int page, int size, Direction direction, S
    this(page, size, new Sort(direction, properties));
}

/**
 * Creates a new {@link PageRequest} with sort parameters app
 *
 * @param page zero-based page index.
 * @param size the size of the page to be returned.
 * @param sort can be {@literal null}.
 */
public PageRequest(int page, int size, Sort sort) {
    super(page, size);
    this.sort = sort;
}

/*
 * (non-Javadoc)
 * @see org.springframework.data.domain.Pageable#getSort()
 */
public Sort getSort() {
    return sort;
}

/*
 * (non-Javadoc)
 * @see org.springframework.data.domain.Pageable#next()
 */
public Pageable next() {
    return new PageRequest(getPageNumber() + 1, getPageSize()
}

/*
 * (non-Javadoc)
 * @see org.springframework.data.domain.AbstractPageRequest#p
 */
public PageRequest previous() {
    return getPageNumber() == 0 ? this : new PageRequest(getP
}

/*
 * (non-Javadoc)
 * @see org.springframework.data.domain.Pageable#first()
 */
public Pageable first() {
    return new PageRequest(0, getPageSize(), getSort());
}

/*
 * (non-Javadoc)
 * @see java.lang.Object#equals(java.lang.Object)
 */
@Override
public boolean equals(final Object obj) {
        if (this == obj) {
            return true;
        }

        if (!(obj instanceof PageRequest)) {
            return false;
        }

        PageRequest that = (PageRequest) obj;

        boolean sortEqual = this.sort == null ? that.sort == null

        return super.equals(that) &amp;&amp; sortEqual;
    }

    /*
     * (non-Javadoc)
     * @see java.lang.Object#hashCode()
     */
    @Override
    public int hashCode() {
        return 31 * super.hashCode() + (null == sort ? 0 : sort.h
    }

    /*
     * (non-Javadoc)
     * @see java.lang.Object#toString()
     */
    @Override
    public String toString() {
        return String.format("Page request [number: %d, size %d,
                sort == null ? null : sort.toString());
    }
}
```

标签： APP Java Hibernate HTTPS Spring Data

评论

快来抢沙发吧！

提交